Modelling EU DIRECTIVE 2016/680 using Enterprise Architecture

A proof of concept for modelling and compliance assessment of the General Data Protection Regulation

CHARLIE PALMÉR
**Abstract:** This master thesis attempted to create an enterprise architecture tool implementation of the new EU legislation General Data Protection Regulation. Using the archimate standard as well as the tool EAAT (Enterprise Architecture Analysis Tool) changes have been proposed so that compliance with the legislation can be verified. 19 of the legislation articles are discussed and implemented with another eight discussed why they are not implemented. Each implemented article is followed by a model snapshot as well as some pseudo-code for easier implementation independent of the tool used. Due to unforeseen problems, no working implementation was not created but the proposed changes can be made to fit any tool built on Archimate’s standard. Finally the proposed modification is discussed and future work related to modelling of the General Data Protection Regulation is proposed.

**Keywords:** Enterprise Architecture, Metamodelling, Legislation Compliance, Compliance Assessment
Table of Content

1 BACKGROUND 5
   1.1 Introduction ................................................. 5
   1.2 General Data Protection Regulation .......................... 5

2 MAIN OBJECTIVE 5
   2.1 Scope and delimitations ..................................... 6
   2.2 Disclaimer .................................................. 6

3 THEORY AND RELATED WORK 6
   3.1 Enterprise Architecture ................................... 6
   3.2 Enterprise Architecture Modelling ............................ 6
   3.3 Related Work ............................................... 7
   3.4 Archimate 2.1 ............................................... 8
      Business Layer .............................................. 10
      Application Layer .......................................... 12
   3.5 Enterprise Architecture Analysis Tool - EAAT ............... 13
   3.6 General Data Protection Regulation .......................... 13

4 METHOD 15

5 ARTICLES 16
   5.1 Breaking down articles .................................... 16
      Compliance .................................................. 17
      Article 5 - Principles relating to personal data processing . 18
      Article 12 - Procedures and mechanisms for exercising the rights of the data subject .......... 20
      Article 13 - Right in relation to recipients .................. 21
      Article 14 - Information to the data subject ................. 22
      Article 15 - Right of access for the data subject ............ 24
      Article 16 - Right to rectification ........................... 24
      Article 17 - Right to be forgotten and to erasure .......... 25
      Article 18 - Right to data portability ....................... 27
      Article 19 - Right to object .................................. 28
      Article 22 - Responsibility of the controller ................. 30
      Article 23 - Data protection by design and by default ......... 30
      Article 28 - Documentation ................................... 31
      Article 30 - Security of processing ........................... 33
      Article 31 - Notification of a personal data breach to the supervisory authority ................. 34
      Article 32 - Communication of a personal data breach to the data subject .................... 35
      Article 33 - Data protection impact assessment ............... 37
      Article 34 - Prior authorisation and prior consultation ....... 38
      Article 35 & 37 - Designation and Tasks of the Data Protection Officer ....................... 39
   5.2 Left out articles ............................................ 41
      Article 6 - Lawfulness of processing ........................ 41
      Article 7 - Conditions of consent ............................ 41
      Article 8 - Processing of personal data of a child ........... 42
      Article 9 - Processing of special categories of personal data ...................................... 42
      Article 11 - Transparent information and communication .... 42
      Article 20 - Measures based on profiling ..................... 42
      Article 26 - Processor ......................................... 42
Article 36 - Position of the data protection officer  

6 IMPLEMENTATION  

7 EXAMPLE  
7.1 As-is  
7.2 Modifications for compliance assessment  

8 USAGE  

9 DISCUSSION  
9.1 Difference between documents  
9.2 Possible benefits  
9.3 Downsides  
9.4 Implementation  
9.5 Ethical Aspects  

10 CONCLUSION  
10.1 Future Work  

Appendices  
A Images
1 BACKGROUND

1.1 Introduction

When it comes to laws and regulations regarding storage and usage of personal data by enterprises the texts are old. The most recent version of regulation set by the European Union was implemented in 1995. A lot has happened since then. In 2011, Apache released their first version of hadoop [5], an open-source framework for distributed processing of large data sets. Since 1995, Cray Inc, a company who focus on processing power, have increased their power by 9,600% [23]. This goes to shows how much the world of IT has changed since last the regulations regarding personal data were changed. So in 2012 a reform of the old directive was proposed by the European Commission, called the General Data Protection Regulation.

1.2 General Data Protection Regulation

The regulation in place from 1995 focuses on a few key legal principles, such as fair and lawful processing, transparency and security among others. The regulation (95/46/EC) established in 95 has been implemented into the laws of the 28 member states. There are however differences in how the regulation is implemented between the different countries. Therefore a new document was proposed to “harmonise the current framework” [3]. The new document would be a legislation, meaning that it is a law that each of the member states have to abide by.

In 2012, the European Commission proposed a new regulation regarding personal data, the General Data Protection Regulation. Since then, the proposal has been worked on by different institutes and departments of the European Union in order to refine the proposal. In order for the regulation to come into effect, the Union has to vote for it, but it is presumed that it will pass.

If the proposal is passed, it will force many companies to change and modify their existing IT architecture so that they comply with the regulation. In order to aid with this transition and to ensure future compliance, this thesis will use Archimate [1] to create a tool usable by enterprises so they can model, validate and test a proposed system to help ensure compliance.

The regulation contains 91 articles. However, all of them are not regulating how an enterprise are to act in accordance with the regulation. This thesis has included 19 of these articles in the proposed model. A model created using the concept of Enterprise Architecture and the tool EAAT [16].

2 MAIN OBJECTIVE

The goal of this thesis is to create an extension to the Enterprise Architecture Analysis Tool [16] in which the requirements presented in the General Data Protection Regulation (GDPR) are implemented. This extension is to be used as a guidance tool to evaluate an enterprises current system and whether or not they are compliant with the different articles presented in the GDPR. The work can be broken down into three parts.

1. Analyse the GDPR and sort out articles not directly relating to a companies interests. For instance, article 48 [10, p. 76] focus on the conditions for the members of the supervisory authority and is not directly affecting the enterprise’s work.

2. Reduce the amount of articles modelled to create a more universally usable model for compliance assessment.
3. Evaluate the proposed model by testing it on an existing model based on a Swedish lottery company as well as interview with a professional within the area of IT security and law.

2.1 Scope and delimitations
The objective is to create a tool usable in as many situations as possible, independent of what kind of information a company is working with. The work is intended to be implemented using EAAT and therefore certain restrictions on how the modelling works are present, such as not to create double models or overly complicated and messy models. In order to comply with this, some of the articles that initially was deemed within the scope of the thesis have been left out. This was done to comply with the idea of not making the model more complex than needed. These articles are discussed briefly in section 5.2.

This thesis is based on the proposal of GDPR [10]. However, during the writing of this document, the European Commission released a new version of the regulation, L:2016:119 [11] which is the finalized legislation. After examining the new legislation there are some differences between them and this is further discussed in section 9.1 page 51. Since the work in this thesis is a proof of concept rather than a functional implementation the work will still be based on the old version.

2.2 Disclaimer
The author of this thesis is in no way a professional within the area of IT law. As such, the interpretations of the articles may be different in a court of law. Therefore the author suggest that the result of this thesis is used in tandem with professionals and not be used as the sole test of compliance.

3 THEORY AND RELATED WORK

3.1 Enterprise Architecture
Enterprise Architecture is described by FEAPo as a well-defined practice for conducting enterprise analysis, design, planning and implementation to ensure a successful development and execution of strategy [7]. Through the use of EA an organisation can evaluate and change their structure in such a way that they achieve their goals while lowering the amount of work required to realise the changes [2]. One of the objectives of Enterprise Architecture is to reduce system complexity and increase value for the organisation when it comes to their IT systems [22].

There are several frameworks for conducting enterprise architecture, such as TOGAF [14], Zachman [24], and Federal Enterprise Architecture [15]. Each framework aim to increase the business value by allowing companies and organisations to keep track of their IT architecture and design them so that they are more efficient.

As mentioned, there are several frameworks available and choosing the right one is no simple task. As Roger Sessions puts it, ”Choosing between Zachman and TOGAF, for example, is like choosing between spinach and hammers.” [20] In his paper, the major differences between the four major frameworks, their strengths and weaknesses are discussed.

3.2 Enterprise Architecture Modelling
There are several frameworks for Enterprise Architecture, TOGAF [14], Zachman [24] and Federal Enterprise Architecture [15] are a few of them.
This paper work with Archimate [1], a modelling language based on the TOGAF framework created by the Open Group. Archimate breaks down an organisation’s structure into four parts, Business Layer, Application Layer, Technology Layer and the Motivational Extension.

- **The Business Layer** is the topmost layer describing the services and products offered, realised by processes performed by actors and roles within the organisation.

- **The Application Layer**, the second layer, is what supports the top layer through software and their services. It also connects the Business Layer with the lowest layer, the technology layer.

- **The Technology Layer** contains the infrastructure of system, such as storage units and network connecting and running the different applications.

The final part of Archimate is the Motivation Extension. This is used to model the motivations, or reasons, for different parts of the structure. Concepts such as Goals or Drivers are modelled using this extension. These concepts may not affect a system directly but will influence the directions and reasons behind certain structural changes within an organisation.

### 3.3 Related Work

The idea of creating a meta-model integration of legal texts is not something new. Waltl et al. propose a solution for compliance to financial laws in regard to IT systems. Their solution is implemented using Archimate and TOGAF. [6] They proposed that concrete legal obligations are modelled using requirements, goals and principles with the goal of ensuring compliance with legal frameworks throughout different phases. However, their work are more focused on legality towards financial systems and are not regarding the hardware solutions as much. As such, it makes it hard to utilise their solution for this thesis.

Furthermore, implementations of risk analysis in Enterprise Architecture Management (EAM) are quite common. Grandry et al. proposed a security risk management using concepts, creating an integration between the IT architecture and the conceptual layer allowing businesses to identify problem areas. [13] One of their goal was to create a model that incorporates information system risk analysis into the Archimate standard.

A problem with using metamodelling for testing compliance is that most EAM systems work on a posteriori, making it hard to develop new IT architecture that comply with the rules and laws set in place. N. Mayer [17] discuss a solution to this problem, creating an ISSRM (Information System Security Risk Management) model focusing on the cost and return on investment of security implementations. His goal is achieve a required level of security with the lowest cost. In the thesis he creates metrics for ISSRM such that as-is and to-be models can be compared.

The work done by Grandry et al. and N. Mayer are both focused on IT security rather than legislative compliance. However, the concepts are similar as there is a need to ensure compliance with set goals and requirements.

Chung et al. discuss non-functional requirements [8] in regards to software engineering. They mention that such requirements are often informal and it is hard to verify that they are fulfilled once the development process ends. However, their proposal of an NFR framework would deal with such issues by creating a way of documenting goals and sub-goals, thus creating a way of ensuring compliance with the requirements. There are several similarities with NFRs and the goal of this thesis such as tracking of goal compliance and the idea of creating sub-goals of the articles in GDPR to break down the requirements. However,

\[1\] NFR’s are a collection name for all requirements that is non-functional such as goals, performance or constraints [9, 12]
the work done by Chung et al. are based not on Archimate but it currently a stand-alone solution. Since one of the goals with this thesis is to create a functioning model using EAAT and Archimate, NFRs are not applicable.

3.4 Archimate 2.1

In section 3.2 Archimate were briefly described. This section contains a deeper explanation of Archimate and some parts of it that will be utilised in this thesis.

As seen previously, Archimate is a modelling framework based on TOGAF. TOGAF aims to break down an architecture into three levels, called layers. These layers contain different concepts within an enterprise and aim to differentiate them. The different layers have set rules on how they interact with each other thus creating a concept where changes in one layer does not have to affect the other layers.

Using the Archimate framework these layers can be visualised and analysed in a structured fashion. The framework are composed of three different parts. First is the Archimate Core containing the different layers. Within each layer are different classes corresponding to a service within that layer as seen in figure 1. Each of these classes have a set function and meaning within the structure.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Business</td>
<td>Business value</td>
<td>Business value</td>
</tr>
<tr>
<td>Business layers</td>
<td>Behavioral concepts</td>
<td>Behavioral concepts</td>
</tr>
<tr>
<td>Business processes</td>
<td>Controlled</td>
<td>Controlled</td>
</tr>
<tr>
<td>Application</td>
<td>Application service</td>
<td>Application service</td>
</tr>
<tr>
<td>Application layers</td>
<td>Application</td>
<td>Application</td>
</tr>
<tr>
<td>Application processes</td>
<td>Controlled</td>
<td>Controlled</td>
</tr>
<tr>
<td>Technology</td>
<td>Technology</td>
<td>Technology</td>
</tr>
<tr>
<td>Network</td>
<td>Network management</td>
<td>Network management</td>
</tr>
<tr>
<td>Communications (as part)</td>
<td>Communication</td>
<td>Communication</td>
</tr>
</tbody>
</table>

Figure 1: The Archimate core with the classes contained in each of the three layers.

Second part is the Extensions, Motivational (see figure 2) and Implementation and Migration (see figure 3). The motivational extension aim to describe the reasons that underlie an enterprise architecture. They describe the motivations behind the structure, such as goals, drivers, and stakeholders. The motivational layer focus less on the actual services and processes of an enterprise but rather on the reasons behind the architecture.

The Implementation and Migration extension adds concepts that are utilised during late stages of the architecture development method (ADM, part of the TOGAF concept). This extension will not be used in this thesis as the extension focus on the implementation of a new architecture or the migration of an existing one into another system.
Finally there are the Relationships connecting the above mentioned classes. As seen in figure 4 there are a few different ones available. In the proposed models (section 5.1) only three of these are used. Firstly, specialisation is used to describe new classes proposed. Secondly the models have combined the use of Association and Access and replaced them with link names. This is how EAAT has implemented the relationships and thus, how the models are made. For clarification, a specialisation is a subclass of another, existing class. It would maintain all the properties of the parent with modified contents. It is used in a more specific case to ensure a clear and understandable implementation. Relations on the other hand are how two classes interact with each other. So a specialisation should have the same relations as its parents class in order to maintain simplicity while offering a more precise model.
In section 4, the articles of the GDPR document will be discussed and models based on Archimate are proposed. In order to better understand each of them, some of the more used classes from the Archimate Core will therefore be explained so that the reader can more easily understand the reasoning behind the design choices. The definitions in section 3.4 below all come from the Archimate 2.1 Specification [1].

**Business Layer**

![Business process diagram](image)

**Figure 5:** Archimate’s defined notation for a Business Process

"A **Business Process** is defined as a behaviour element that groups behaviour based on an ordering of activities. It is intended to produce a defined set of products or business services."

Each business process represents actions taken by a company in order to create a service or product. These processes can be connected to each other, where the next one use what the previous created and adds to it, delivering a new product or service to the next step.

![Business service diagram](image)

**Figure 6:** Archimate’s defined notation for a Business Service

"A **Business Service** is defined as a service that fulfils a business need for a customer (internal or external to the organization)."

A business service is used to model a point in the architecture where a finished product is...
delivered. They are created by processes, functions or collaborations and can symbolise the end of a processing chain. It signifies that some sort of product or service is delivered to an external role.

![Business Event](image)

**Figure 7:** Archimate’s defined notation for a Business Event

"A **Business Event** is defined as something that happens (internally or externally) and influences behaviour."
The business event is used as a trigger for certain processes or other business behaviour, such as a customer requesting some product information or making an order for a service thus starting the process chain.

![Business Object](image)

**Figure 8:** Archimate’s defined notation for a Business Object

"A **Business Object** is defined as a passive element that has relevance from a business perspective."
The idea of business objects is to model informational or conceptual parts of a business. These could be an object type such as *Invoice* or it can represent actual information created or used by the business layer.

![Representation](image)

**Figure 9:** Archimate’s defined notation for a Representation

"A **Representation** is defined as a perceptible form of the information carried by a business object."
A representation is a piece of information related to business objects. They are used to model actual pieces of information used and created by the business layer.

![Business Actor](image)

**Figure 10:** Archimate’s defined notation for a Business Actor

"A **Business Actor** is defined as an organizational entity that is capable of performing behaviour."
Business Actors represent an organisational unit, meaning they only exist in the business layer. The actor can represent a single human or several, a department or a business unit. They can be assigned to business roles. In this thesis, the business actor is used to represent external units, such as customers.

Figure 11: Archimate’s defined notation for a Business Role

"A Business Role is defined as the responsibility for performing specific behaviour, to which an actor can be assigned."
The business role represent the person or department in charge of the behaviour elements of the business layer. The roles are connected to one or several business processes and represent that the role is the one performing the process or service.

Application Layer

Figure 12: Archimate’s defined notation for an Application Component

"An Application Component is defined as a modular, deployable, and replaceable part of a software system that encapsulates its behaviour and data and exposes these through a set of interfaces."
An application component is used to model the existence of a system. A system that delivers a number of specific application services to the architecture. Simply put, an application component is a software used by the architecture.

Figure 13: Archimate’s defined notation for an Application Function

"An Application Function is defined as a behaviour element that groups automated behaviour that can be performed by an application component."
An application function describe the internal functions of an application component. The function is access externally by Application Services and each application function represent a single function within the application component such as Saving or Billing.

Figure 14: Archimate’s defined notation for an Application Service
"An **Application Service** is defined as a service that exposes automated behaviour."
The application service, as mentioned above, is what exposes an application function to the environment. These represent the action made by a component such as *Bill creation* or *Transaction Processing*. The application services are one of the connections between the business and application layer as it can be accessed by i.e. business processes.

![Data object](image)

**Figure 15:** Archimate’s defined notation for an Data Object

"A **Data Object** is defined as a passive element suitable for automated processing."
A data object represent data utilised by application components and are accessed by application functions. They are created, modified and so on at the application layer. They can also realise Business Objects. This realisation is used throughout this thesis but is not shown as the Data Objects are connected directly to the business layer. This is done since modifications to a data object require the interaction of an application function. Since the proposed models are built to be utilised in any environment regardless of current application layer architecture, the bypass has been implemented to reduce the pre-existing requirements on existing enterprise architectures.

### 3.5 Enterprise Architecture Analysis Tool - EAAT

The **Enterprise Architecture Analysis Tool (EAAT)** is a software suite that utilises the Archimate framework to model an enterprise architecture. EAAT was created by P. Johnson et al. with the intention of creating a tool that can provide a costs analysis as well as system quality analysis [16].

EAAT is built using the *Predictive, Probabilistic Architecture Modelling Framework (P²AMF)*, a framework for calculating probabilistic variables of an enterprise architecture. This allows the user to enter not a set evidence but one that varies. This allows for more accurate models of the real world thus making the decision making based on the models more reliable.

During this master thesis, EAAT was used for implementing the models. Not because its ability to calculate probabilistic results, but since it was the tool the author was most familiar with. All models shown below are made using EAAT.

### 3.6 General Data Protection Regulation

During the past years, data breaches have become more and more common, and with each breach, more personal data is leaked. Looking at Statista’s information regarding data record compromises (figure [16] the numbers are staggering.
In 2012 the European Commission proposed a new regulation that would modernise the laws regarding personal data within the EU. Up till that point, the old rules were just a directive, meaning that each member state would create their own laws following the directive. Each of the 28 member states have implemented the directive, but there are differences in how they are enforced.

So with the new regulation, the European Commission proposed it would be a legislation. This would make it so that all member states have the same laws. The goal was to unify and strengthen data protection as well as creating a simpler way for international companies to work within the EU. Thirdly they aimed to increase security when exporting data from the EU. Lastly, they wanted to increase the security for individual EU citizens by requiring companies to obtain consent from said citizen. They also shifted the burden of proof from the individual towards the organisations, increasing the strength for a single citizen.

The proposal were sent back and forth between the different instances of the European Commission, and in May of 2016, the legislation was passed.

With a two year adoption period, companies and organisations within the EU must modify their organisation in order to comply with the new legislation. However, the legislation also affects any company or organisation that is external to the EU but conducts business within any member state of the EU.

Failure to comply with the regulation can be expensive. There are two types of fines described within the regulation. First one is 10 000 000 Euro or up to 2% of annual worldwide turnover. Second one is double the first, 20 000 000 Euro or 4% of annual worldwide turnover. In both cases, the highest value is chosen. These fines are substantial, showing that the EU are motivated to ensure the data security of its members.
There are some exceptions to who is affected. Law enforcement and matter of national security as well as any data storage for personal use are excepted [11, p.32 Article 2] and some of the articles are lenient on smaller companies and organisations.

The legislation aim to put several measures in place to reduce the risk of unlawful handling of personal data and increase the rights of the data subjects. The data subjects will have the right to launch complaints and request data deletion, modification and migration. The companies must comply with these requests or have sufficient reasons why they do not, and the burden of proof lies with the companies. Companies are also required to appoint a Data Protection Officer (discussed in section [5.1, page 39]) who is responsible for contact with the authorities as well as ensuring that the company are compliant with the regulation.

4 METHOD

In order to create a meta model for GDPR compliance assessment, all of the articles of the regulation had to be assessed. Each of the articles of the document were read, after which they were kept or discarded, based on their direct impact for a company. For instance, article 70, Tasks of the chair is in regards to the proposed European Data Protection Board, a new division within the European Union. It will not affect companies in a direct manner and thus, it is discarded from the scope of this thesis.

After the initial read-through, the kept articles were re-read and a new selection took place, this time focusing on articles pertinent to special cases relating to companies (discussed in section [5.2, page 41]).

This lead to 23 articles out of 91 being deemed directly affecting companies. The next step would be to map the legislative text into the Archimate framework. The goal was to create a meta model similar to the one available in EAAT so that adoption into existing models would be easier. In section [5.1] each of the chosen articles are listed and each part is broken down into three parts, Motivation, Model, and Pseudo-code. Motivation describe the article and proposes a solution for mapping it onto the Archimate framework as well as any needed modifications to said framework. The model section simply contains, where applicable, a model of the described modifications, granting the reader a visual aid in understanding the proposed changes. Finally, the Pseudo-code section contains some basic code in how the model would validate and assess compliance with the regulation as well as any other code required for this assessment.

During this part of the work, several new specialisations and relations are being proposed in order to create a functioning compliance assessment tool. Specialisations are proposed so that the overall functionality of the superclass is maintained and allows EAAT to still evaluate them as normal. For clarification, a superclass is the parent of a specialisation. In our case, a proposed change to a class already existing in Archimate would make that pre-existing class a superclass. Specialisations are primarily used for Business Process. Each case is described in section [5.1] but in general the reasoning behind this is because there is a need to differentiate between processes for compliance assessment.

In cases where relations have been used instead or together with specialisations the reasoning has been to clarify the intention of the relation and to allow some connections that are unique for the specialisation (i.e. not part of the superclass).

After the modifications were theorised, an implementation into EAAT were attempted. However, due to unforeseen issues, further discussed in section [6, page 43], no real working implementation was created. All the specialisations and relations were created but due to issues with naming of relations as well as the authors lack of knowledge of EAAT, the implementation was scrapped for time.

Finally, the modified meta-model exemplified using a model created by the author for a different project. This project were connected to a large company in Sweden and as such, any information as to which company it might be has been censored. The original model
is discussed followed by a discussion on modifications needed in order to conduct a GDPR compliance assessment using the modified meta-model. However, since no fully working implementation of the modifications exist, the compliance assessment were done outside of EAAT.

5 ARTICLES

As mentioned in section 4, each of the 23 articles that were kept are discussed below. Following that, eight articles that initially were kept but later scrapped due to being applicable to special cases are discussed.

Firstly some explanations of assumptions are made so that each of the models are easier to understand are made. Following that, the overall compliance test is described and shown, followed by each of the articles. Finally the reasons for discarding some articles are presented.

5.1 Breaking down articles

In this section each of the articles that have been chosen to be evaluated and modelled are described. They are broken down and the reasoning behind the modelling is given for each article. Some assumptions are made to reduce the complexity of the models.

First assumption is based on the way Archimate models data objects. A data object is described as "Typical examples of data objects are a customer record, a client database, or an insurance claim". [1, sec. 4.4.1]. As can be seen it can be a single unit of data or a collection of similar data units, such as a database. Both cases are used in the proposed models below and as such, there is a need to be able to tell the difference between the different uses. So to illustrate this, a relation called regards (shown in figure 17) has been introduced. It symbolises that the data object only contain data pertinent to a single person and not all data from the storage unit.

Looking at image 17 there is a unit called GDPRBusinessActorDataSubject connected to the class PersonalDataObject. Both of these are specialisations of Business Actor and Data Object respectively. Since the model contains both, the data represented by PersonalDataObject only contain data relating to a single person, the data subject. In order to show this, the relation regards has been proposed.

However, in order to decrease the amount of modelling, whenever a model found below that model both the data storage and a specialisation of the class Business Actor the new relation is implied but not shown in the models.

Second assumption is that several articles calls for the Controller to perform certain actions. Article 4(5) [10, p. 41] states that a controller is ".. which alone or jointly with others determines the purposes, conditions and means of the processing of personal data; ..". For the purpose of this thesis this is assumed to be a representation of the enterprise for whom the model is made. Thus, the controller is, at the highest level, the person who is responsible for all operative decisions, the CEO and/or chairman of the board. Since there is, in most cases, a clear hierarchy within a company modelling a controller is not relevant.
Figure 17: The relation \textit{regards} is assumed in all the below mentioned articles

Compliance

In order to test if the model complies with the General Data Protection Regulation (GDPR) or not, this thesis propose that a specialisation of the Motivation Extension’s \textit{Goal} [1], called \textit{GDPR Compliance} is added (figure 18).

![Diagram](image1)

\textbf{Figure 18: Specialisation of Goal}

This class will be used to verify that each of the below mentioned articles have their requirements fulfilled. In order to test this, several new specialisations are proposed, one for each of the below mentioned articles, called \textit{GDPR XX}, where XX is the number relating to the article (see figure 19). Each of these are related to \textit{Requirement} and with them, the required relations will have to be created. If all of the \textit{GDPR XX} classes return True, the model will return that the current structure are in compliance with articles mentioned below. See figure 20 for an overview of this.

![Diagram](image2)

\textbf{Figure 19: Specialisation of Requirement}
Model

![Diagram](image.png)

Figure 20: Proposed model for compliance with article 30

**Article 5 - Principles relating to personal data processing**

**Motivation**

Article five is in relation to the requirement on personal data. It states that all data must be processed lawfully and transparent in regards to the data subject. It states that no unnecessary data is to be collected, meaning all data collected must have a purpose.

§5(a) states that any data should be "processed lawfully, fairly and in a transparent manner in relation to the data subject"). This is further specified in several articles, for instance, article 14 (page 22) and article 15 (page 24). As this statement is fairly ambiguous, it is better dealt with in the other articles mentioned.

Paragraphs 5(b), (c) are somewhat controlled by the document required by article 28 (page 31). These three paragraphs are also written in such a way that the author interpret them as being more conceptual rather than affecting the architecture. So in order to reduce the check-list like implementation and double modelling, paragraphs (a), (b) and (c) of article 5 will be ignored.

However, article 5(d,e) are of interest. They are in relation to the accuracy as well as how long personal data are relevant. If data is deemed irrelevant, there is no need to keep it and should be removed or updated. In order to comply with this article, there is a need for regular data review. The requirement is only relevant for any data storage containing personal data. To distinguish any DataObject from the ones containing personal data, a new class is proposed called PersonalDataObject, seen in figure 21. This class is a subclass to DataObject that should be used whenever a data object contains personal information.

![Diagram](image.png)

Figure 21: Personal Data Object is a specialisation of the class Data Object

In order to deal with the requirement of regular reviews, the new class contains three attributes. The first one will be a numeric entry called lastReviewDate containing the Julian Calendar date for when a review of the data storage represented was last performed. Second attribute, reviewFrequency represent the amount of days between two reviews. It will have to
be evaluated for each data storage based on how long the data contained will be relevant and accurate. The third attribute, a derived one, will be called \textit{reviewNeeded} used to evaluate if there is a need for a review of the data storage. This operator uses the two previously mentioned attributes as well as an attribute found in the class \textit{GDPR 5} called \textit{dayOfYear} which is the current date in the Julian Calendar. Based on this, the model can evaluate if there is a need for a \textit{DataObject} to have its data reviewed.

\textbf{Model}

Figure 22 show the proposed model for testing article 5 as well as allowing the model to test for overall compliance.

\begin{center}
\includegraphics[width=0.5\textwidth]{model.png}
\end{center}

\textbf{Figure 22}: Proposed model for compliance with article 5

\textbf{Pseudo-code}

\begin{verbatim}
GDPR_5:
    if any(PersonalDataObject.reviewNeeded) = True
        return False
    else
        True

The code for testing the need of data review (shown below) is based on the Gregorian calendar and assumes that there are some kind of built-in date-functionality.

reviewNeeded:
    var daysSinceLastReview = gdpr_51.today() - lastReviewDate
    if daysSinceLastReview >= reviewFrequency
        return True
    else
        return False
\end{verbatim}
Article 12 - Procedures and mechanisms for exercising the rights of the data subject

Motivation

§1 states that the controller should be able to prove that measures have been taken to ensure compliance with article 13 and articles 15 to 19. Each of them are discussed in their own sections below and as such, the work here is used to verify compliance with this paragraph.

§2 require the controller to inform a data subject who has submitted a request concerning article 13 or articles 15 to 19 if any actions have been taken in accordance with the request. If the controller refuses to take action, the controller shall inform the subject with the reasons for the refusal as well as inform them about their right to lodge a complaint to the supervisory authority. In figure 24 the representation would contain this information and the business service represent that the company makes the information available to the data subject.

§1 will check each of the new classes GDPRXX (from articles 13 and 15 to 19), if any of them are returning the value False, article 12 can not be considered to be in compliance. This is shown in figure 23. The second part of the article require that a request pursuant to the before mentioned articles be answered by the controller. In order to test this, a new class is proposed called GDPRRequestResponse, a specialisation of BusinessProcess. This class will be connected to a BusinessEvent representing any request made in accordance with the mentioned articles. The new class will create a representation containing any and all information required to fulfil the requirements of the article 12. This representation is what is sent to the data subject, represented by the proposed class GDPRBusinessActor-DataSubject further described in article 18 (page 27). The complete model can be seen in figure 24 and pseudo-code for the check is found on page 21.

Model

![Proposed model for compliance with article 12.1](image-url)
Figure 24: Proposed model for compliance with article 12.2

Pseudo-code

GDPR_12.1:
if GDPRGoalCompliance.GDPR13 and ... and ...GDPR19 then
    True
else
    False

GDPR_12.2:
if GDPR_12.1 then
        True
    else
        False

Article 13 - Right in relation to recipients

Motivation

If the controller rectifies or erases any personal data in accordance with articles 16 (page 24) or article 17 (page 25), each recipient whom the data have been disclosed to shall be informed of the changes in the data unless this proves impossible or involves a disproportionate effort.

Easiest way to model this is by adding a information step to each of the articles. In order to accomplish this, a new class is proposed called GDPRInform which represent the step of
informing a third party to whom any personal data have been disclosed. The new class is a specialisation of Business Process as seen in figure 25.

Figure 25: GDPR Inform is a specialisation of Business Process

Model

Figure 26: Proposed model for compliance with article 13

Pseudo-code

GDPR_13:
    if Exists(PersonalDataObject.read.GDPRInform.Informs.BusinessRole) then True
    else False
    else True

Article 14 - Information to the data subject

Motivation

Whenever personal data is collected, the controller has a responsibility to inform the data subject about it. §1 states what information is to be given the data subject. §2 and §3 add, depending on the situation, more information the data subject should receive from the controller. §4 states the time frame for when the information should be delivered to the data subject. §5 states some exceptions for when contacting a data subject is not necessary.
In order to test this article, a new class is proposed called *GDPRDataCollection*, another specialisation of *BusinessProcess*. This class will have a write-relation to one or more *PersonalDataObjects* as well as a write-relation with a *Representation*, containing the information mentioned in §1 and situational information from §2 and §3. The *Representation* is connected to a *BusinessService* representing sending the information to the data subject. The data subject is represented with the new class *GDPRBusinessActorDataSubject* described in article 18 (page 27).

This means that *GDPRDataCollection* will write data into the data storage (PersonalDataObject) and by doing this, it will also need to create a document informing the data subject of such actions, as seen in figure 27.

The new class *GDPRDataCollection* contains an attribute called *Exception*, a simple binary value set to one if any of the exceptions mentioned in §5 are in effect.

**Model**

![Diagram](image)

**Figure 27**: Proposed model for compliance with article 14

**Pseudo-code**

```plaintext
GDPR_14:
    if DataObject.write.GDPRDataCollection.Exception = 0 then
        if Exists(DataObject.write.GDPRDataCollection.write.Representation.read.BusinessService.
            GDPRBusinessActorDataSubject) AND Exists(...BusinessService.
                GDPRDataCollection) then
            True
        else
            False
    else
        True
```

23
Article 15 - Right of access for the data subject

Motivation

A data subject should be able to, on request, obtain information if personal data relating to them are being processed. If that is the case, §1 states the content of the information the subject are entitled to. A data subject shall also have the right to obtain communication of the personal data undergoing processing.

In order to test compliance, a specialisation of Business Process, GDPRProcessingInquiry, is proposed. This should represent that an enterprise have methods and procedures in place when and if a data subject makes such a request.

The model will also use the class GDPRBusinessActorDataSubject, described in article 18 (page 27), to create a loop used for model verification.

Model

Figure 28: Proposed model for compliance with article 15

Pseudo-code

GDPR_15:

then
    True
else
    False

Article 16 - Right to rectification

Motivation

If a data subject deems any of their personal data incorrect or incomplete, they have the right to request a rectification of the data, including supplying the correct information.
To model this, a new class is added, \textit{GDPRDataRectification}, a specialisation of \textit{BusinessProcess}, representing the change of personal data. To check if the architecture is in compliance, the new class will require a read as well as a write-relation.

\textbf{Model}

\begin{center}
\includegraphics[width=0.5\textwidth]{figure29.png}
\end{center}

\textit{Figure 29:} Proposed model for compliance with article 16

\textbf{Pseudo-code}

\begin{verbatim}
GDPR_16: if Exists (PersonalDataObject.Writes .
           GDPRDataRectification.Read.Representation) then
          True
else
          False
\end{verbatim}

\textbf{Article 17 - Right to be forgotten and to erasure}

\textbf{Motivation}

A data subject shall have the right to, at any time, request that any processing using their data stop. They can also request that the data be removed from the systems and any dissemination stops. §1 states the grounds for which such a request is legal.

§3 states some exceptions to the need for erasure, such as for reasons of public interest in the area of public health or in special cases referred to in §4 of article 17. Instead of erasure, the controller may restrict processing of personal data if any of the criteria are fulfilled. If §4 is in effect the controller are not allowed to process the data unless it it fulfils one or more of the exceptions stated in §5. §6 states that the subject should be informed before the restrictions set in due to §4 are lifted.

§7 states that there should be measures in place so that out of date data is reviewed and removed and/or updated. Such measures are introduced in article 5 previously mentioned (page 18)

Since article 17 contains many situations, exceptions and exceptions to the exceptions, it is near impossible to create a model that is simple to use yet covers all the requirements. Each request will most likely be dealt with on a case by case basis and as such, each outcome and the reasons behind will probably be unique. Thus this thesis propose that the model is
made as simple as possible, instigating a manual check for each request received.

To achieve this, it is proposed that a new class, *GDPR Erasure*, shown in figure 30, a specialisation of *Business Event*, is added and should represent the need for a manual review of the erasure request. So whenever a request pursuant to article 17 is received, the controller will have to check the data and evaluate if the request is legit or not and take the appropriate measures described in the article.

![Diagram](image)

**Figure 30:** GDPR Event is a specialisation of the class Business Event

**Model**

![Diagram](image)

**Figure 31:** Proposed model for compliance with article 17

**Pseudo-code**

The code will check that the loop from the request to limit and/or stop processing of personal data is received until an answer has been given to the data subject. The relations read and write (see figure 31) between *Request Evaluation* and *PersonalDataObject* have been used since the request is to stop processing. This would result in editing of the data storage unless the request is denied, which here is considered an exception.

```python
GDPR_17:
    if Exists(
        GDPRBusinessActorDataSubject . Owns .
        PersonalDataSubject
    )
```

26
Article 18 - Right to data portability

Motivation

Assuming most of the data an enterprise collects of a personal nature would probably be stored in a SQL Database [21] or any other system for handling large amounts of structured data. Based on this, a data subject shall have the right to get information regarding them in a format such that the data subject can use the data as she please.

If the data has been provided by the subject and the processing is based on a contract or consent, the data subject shall have the right to request that the data is transferred into another data storage system without any hindrance from the controller.

Basically, a data subject can request that their personal data is being transferred to a location of the subjects desire. However, there are two states for which the result are different. Thus the model has to check for both situations. First one has to handle requests where the data is not provided or processing based on a contract or consent. In this case, the model will check for a GDPRBusinessEvent, ensure that there is a write-relation into a new Data Object representing the personal data being transferred and end with a Business Service connected to the new class GDPRBusinessActorDataSubject whom should also be connected to the GDPRBusinessEvent since the data subject would be the owner of the request.

The second part is similar but asks for a contract or consent between the controller and the data subject to be established prior to the data collection. Simplest way to incorporate this into the model without having to heavily modify Archimate’s definition of Contract would be to add it as a attribute in the class PersonalDataObject. With this value set to True the model can test whether or not the data can be requested to be moved into another, external system.
Model

![Proposed model for compliance with article 18](image)

**Figure 32:** Proposed model for compliance with article 18. Note the attribute in *PersonalDataObject*

**Pseudo-code**

GDPR\_18.1:

```java
if Exists (
    BusinessService.GDPRBusinessActorDataSubject.
    PersonalDataObject)
    True
else
    False
```

GDPR\_18.2

```java
if GDPR\_18.1 AND
    PersonalDataObject.ContractConsent = True then
    True
else
    False
```

**Article 19 - Right to object**

**Motivation**

A data subject can object to the use of their personal data if the reasons for processing is based on paragraphs 6:1(d,e,f) [10, p.44]. It is then up to the enterprise to demonstrate compelling reasons to override the objection.

If the processing of personal data is directly used for marketing purposes, a data subject should be able to object, free of charge, to the usage of their information. This right have to be clearly stated, discernible from other information.
The first paragraph is dependent on the reasoning behind the processing of personal data. Depending on the reasons, they might be stated in the processing document in accordance with article 28:2 (page 31). However, it could be so that the reasons are not stated and so to ensure that the model is applicable in as many situations as possible, it is proposed that the objection is triggered using the new class GDPRBusinessEvent thus giving the model a chance to test that the procedures are in place to handle such an event.

The model will check that a certain loop exists, but ignore if the data is used for marketing or fall under article 6:1(d,e,f). The loop goes from the GDPR 19 up through the data object containing personal data, through a read-relation to a business process. From here it goes back down to the data object using a write-relation ending back at the GDPR 19-class. This ensures that any changes needed to the usage of personal data is edited in the storage unit.

**Model**

![Proposed model for compliance with article 19](image)

**Pseudo-code**

GDPR_19:

```
if DataObject.PersonalData then
    if Exists (DataObject.BusinessProcess.
        GDPRBusinessEvent)
    AND Exists
        (DataObject.read.BusinessProcess.write.
        DataObject) then
        True
    else
        False
else
```

29
Article 22 - Responsibility of the controller

Motivation

§1 of article 22 states that the controller shall ensure that any processing of personal data is in compliance with the regulation. Basically, this is what is trying to be achieved in this thesis using EAAT [16]. The overall goal is called GDPR Compliance (page 17) and as such, if that is fulfilled, so is §1. So there is no need to additional modelling here.

§2 states that the controller shall take additional steps to ensure that articles 28, 30, 33, 34 as well as 35 are in compliance with the regulation. All of these are being handled in individual sections as part of the thesis. Thus, no additional modelling is needed. If the entire model is in compliance, this will, by default, be fulfilled.

Article 23 - Data protection by design and by default

Motivation

§1 and §2 of article 23 are of interest in this report. They both are in regard to ensuring that the personal data is secure. This is, as the title of this section states, done in two ways, by design and by default. The design part is stated in §1. It states that having regard to the state of the art as well as cost of implementation, the controller has to ensure that, at all times, the data is secured. They have to make technical as well as organisational measures to ensure compliance with the regulation. This is quite broad and makes modelling it near impossible since it states no direct need on the architecture. However, article 33 (page 37) and article 34 (page 38) both target the overall technical and organisational structure required. Thus, it is proposed that if the model find articles 33 and 34 are in compliance, article 23(1) should also be seen as in compliance with the regulation.

The second paragraph states that, by default, only the data needed to complete a certain task is to be used, as well as not collecting nor storing data for longer than needed. Article 5 [10] p. 43 handles the time of storage, thus no need to do it here as well. §2 also states that only a finite number of people should have access to the data. Being finite is not defined and thus, knowing how many people have access would be sufficient. There is a need for access restriction but, for modelling purposes, it will not state who will have access, just that there are mechanisms in place to handle it.

Depending on how the data storage is implemented, access restriction can be done in several ways and in different layers. It can be modelled at the infrastructure layer or in the application layer. For purpose of this thesis, it will be seen as an application layer implementation.

There are two ways of going about this (see figure 34). The first way is to assume that the access control is done between the application and the data storage. The control lies in the implementation of the data storage. An example of this is how MySQL implements access control [18 sec. 7.2]. There are some limitations to this system but it represent one way of access control. This version of the model is shown in figure 34b below.

It is assumed that few companies work directly in a SQL database (or the like) but rather have the access control in the application. This kind of access control is assumed to be the one most commonly used in an Enterprise environment and as such, is the one used in this thesis, shown in figure 34a.

The implementation of this require the creation of GDPRAccessRestriction, a specialisation of ApplicationFunction. It is in all manners similar to ApplicationFunction but will
be used to test for access control between an *ApplicationComponent* and other parts of the architecture. If a *PersonalDataObject* connected to an Application Component that in turn is connected to an instance of the new class, there should be compliance.

**Model**

![Diagram](image)

(a) Access Control within the application component

(b) Access Control between application component and data storage

**Figure 34:** Two versions of access control

**Pseudo-code**

```plaintext
GDPR_23.2:
    if Exists(PersonalDataObject.ApplicationFunction.ApplicationComponent.GDPRAccessRestriction) then
        True
    else
        False
```

**Article 28 - Documentation**

**Motivation**

Article 28 require the controller, processor and/or the controller’s representatives to document all processing under their responsibility. §2 regards the content of the document while §3 require that the documents are made available to the supervisory authority. §4 contain some exceptions to the requirement. Any natural person whom process personal data without commercial gains are excluded as well as "an enterprise or organisation that employ fewer than 250 persons that is processing personal data only as an activity ancillary to its main activities." [10, p. 59].
As mentioned above, the content of the documents are stated in §2. These will not be added to the model as attributes but will be mentioned in the context help found in EAAT [16].

Since the article states that all processing require documentation it is interpreted such that each unique instance of personal data processing require a document, creating several instances. Therefore, each unique instance of Personal Data Object must be connected to a Representation. In order to distinguish the document required by article 28, a new class is be added, called GDPRProcessingDocumentation, which is a specialisation of Representation. In order to reduce the complexity of the model and following the Archimate 2.1. standard the model will employ the following definition: "a single representation can realize one or more specific business objects.” thus resulting in a single instance of the new sub-class. However, the new requirement GDPR 28 will test that each PersonalDataObject are connected to a GDPRProcessingDocumentation via a write-relation.

Lastly, in order to comply with §3 of article 28, there is a need to have a read-relation connecting the document with the supervisory authority. In article 32 (page 35) a specialisation of Business Role is proposed called GDPRBusinessRoleSA. Using this new class and ensuring that the GDPRProcessingDocumentation class is connected using a read-relation, compliance will be ensured.

**Model**

![Diagram](image)

**Figure 35:** Proposed model for compliance with article 28.1
Figure 36: Proposed model for compliance with article 28.3. Business Process represents any process which are processing personal data.

Pseudo-code

GDPR_28.1:

if DataObject.PersonalData then
  if Exists
    (DataObject.write.
      GDPRProcessingDocumentation) then
    True
  else
    False
else
  False

GDPR_28.3:

if DataObject.PersonalData then
  if Exists
    (DataObject.read.BusinessProcess.read.
      GDPRProcessingDocumentation).
  AND Exists
    (DataObject.BusinessProcess.
      GDPRBusinessRoleSA) then
    True
  else
    False
else
  False

Article 30 - Security of processing

Motivation

Article 30 require the controller and processor to make an assessment of the risks in regard to storing and processing personal data such that it is not deleted, disclosed, changed or
used for other than the intended processing. These risks have to be dealt with, both on a technical and an organisational level. Thus, combining Archimate 2.1’s motivation standard with the technical models, the assessment can be tested.

Adding two specialisations to Requirement, GDPRTechRequirement and GDPROrganisationRequirement, that connect to the motivation class Goal there is a way to connect the assessment with the implementations of the solutions thus creating a way to ensure compliance. The GDPRTechRequirement would deal with the security of the actual data, such as access restriction and backup of the personal data while GDPROrganisationRequirement is in regard to the reasons for processing the personal data as well as who should have access to the data.

Model

![Proposed model for compliance with article 30](image)

Figure 37: Proposed model for compliance with article 30

Pseudo-code

GDPR_30:
if Exists (PersonalDataObject.GDPRTechRequirement)
GDPROrganisationRequirement) then
    True
else
    False

Article 31 - Notification of a personal data breach to the supervisory authority

Motivation

If the enterprise suffer from a data breach in which personal data has been leaked, the regulation require the controller, within 24 hours of noticing the breach, to contact the supervisory authority of the breach. In the same fashion as in article 28 §2, (page 31) the content of the notification is stated in §3 but are left out in the model. However, they will be part of the context help within EAAT [16].
In order to fully comply with this article, the controller is required to document all personal data breaches, collecting the facts about the breach, the actions taken and the effects of the breach so as to let the supervisory authority verify that the actions are in compliance with this article.

So to model this, a new specialisation of *Representation* has been added called *GDPRBreachDocumentation*. This is used to test whether or not the architecture is compliant with article 31.A.

**Model**

![Proposed model for compliance with article 31](image)

**Figure 38:** Proposed model for compliance with article 31

**Pseudo-code**

```plaintext
GDPR_31:
if Exists(
    PersonalDataObject.AssessDataBreach.GDREventDataBreach)
    then if Exists(
        GDPRBreachDocumentation.Read.
        InformSupervisoryAuthority.Informs.
        GDPRBusinessRoleSupervisoryAuthority)
        then True
        else False
    else True
```

**Article 32 - Communication of a personal data breach to the data subject**

**Motivation**

Following article 31 (page 34) if the breach could result in a risk to the privacy or protection of the subject’s data, the controller shall inform the data subject about the breach.

However, §3 states some exceptions for when informing the data subjects are not necessary, such that the controller can show the Supervisory Authority that sufficient technological measures have been implemented to protect the personal data. These measures shall render
§4 states that the Supervisory Authority can require the controller to contact the data subject even if the previous paragraphs are in affect.

The modelling of this article would follow the model from article 31 where the GDPRBusinessActorSA would be the owner of a Business Event, creating a representation that would require the controller to contact the data subject and send the data breach information to them using the class GDPRBusinessActorDataSubject added from article 18 (page 27). The model is based on the idea that the enterprise won’t have to contact the data subject unless the Supervisory Authority require them to. If this is the case, there is currently no model for it, but a company should ensure that they atleast have the procedures stated in case they need to use it.

Model

![Proposed model for compliance with article 32](image)

**Figure 39:** Proposed model for compliance with article 32

**Pseudo-code**

```markdown
GDPR32:
if Exists (DataObject.BusinessObject.RequireInformSubjects.GDPRBusinessActorSA) then
    if Exists (DataObject.BusinessObject.RequireInformSubjects.write.Representation.ContactAffectedSubjects.GDPRBusinessActorDataSubject) then
        True
    else
        False
else
    True (since we might still have to contact
```

the data unintelligible.
Article 33 - Data protection impact assessment

Motivation

Article 33 require the controller or the processor to perform a risk assessment of certain processing operations. §1 require that they are done if the operation presents specific risks to the rights and freedoms of data subjects by virtue of their nature, their scope or their purpose. §2 states some of the operations that would require such an assessment, but it is not limited to just these.

Using the motivation extension to Archimate 2.1 [1] there is a way to model the assessment. However, connecting it to the technical model would be a stretch since, according to the standard, one would have to connect them using the Stakeholder < – > Business Role relation. In order to better connect the assessment with the technical model, this thesis proposes that a new relation is created between Assessment and the newly created class GDPRRepresentation so that a result of the assessment can be used in the technical model. This Representation is readable by Business Process’s so that the changes required can be implemented.

How this article is implemented is, in many ways, based on the specific purpose of data processing operations as well as the goal of the enterprise. Therefore, the article is implemented as bare as possible so that it can be applied in as many situations as possible. This since the assessment done in compliance with article 33 will be used in article 34.

Model

![Diagram](image)

**Figure 40:** Proposed model for compliance with article 33

Pseudo-code
GDPR33:

```python
if exists(ProcessingRiskAssessment.GDPRRiskAssessment)
AND exists(ProcessingRiskAssessment.SensitiveDataProcessing)
    True
else
    False
```

**Article 34 - Prior authorisation and prior consultation**

**Motivation**

Article 34 states situations for which there is a need to acquire prior authorisation from the Supervisory Authority (SA) in order for the processing to be legal. Among the reasons are if the assessment required from article 33 [10, p. 62] states that there is an increased risk or if the Supervisory Authority, due to the virtue of the scope, nature and/or the purpose, deems the specific processing in need of prior authorisation.

In order to acquire prior authorisation, the controller and/or processor shall share the assessment from article 33 with the SA. If the SA returns with a negative answer, they shall also reply with proposals to remedy any non compliances.

§4 states that the SA should compile a list of all processing operations that require prior authorisation, regardless of what the results of the assessment from article 33 are.

§6 asks that all assessments from article 33 are shared with the SA, as well as any other data that might be relevant, in order for the SA to make informed decisions on what processing operations would require prior authorisation.

With this in mind, the following model changes are proposed in order to ensure compliance with article 34.

Input into the Business Process that would process personal data would be the Assessment created in accordance with article 33 as well as a Representation containing the list that is compiled by the supervisory authority. This leads to a junction [1, section 7.3.2] where the options are Prior authorisation required or No authorisation required. If there is a need for prior authorisation, the assessment from article 33 is to be sent to the supervisory authority. The response would be another junction with the options Authorisation granted in which case, the processing can proceed or the response is Authorisation Rejected with the addition of a Representation containing the proposed changes in order to comply with the regulation. From this point, it is assumed that if these changes are implemented, the processing can continue from the new state.
Model

![Archimate diagram showing compliance model for article 34]

**Figure 41:** Proposed model for compliance with article 34

### Pseudo-code

```plaintext
GDPR34:
    if notExists(ProcessingPersonalData.SALLListOfProcess)
    OR notExists(ProcessingPersonalData.GDPRRiskAssessment)
        if notExists(ProcessingPersonalData.RequestAuthorisation.
            GDPRBusinessRoleSupervisoryAuthority.
            AuthorisationGranted)
            if exists(ProcessingPersonalData.
                RequestAuthorisation.
                GDPRBusinessRoleSupervisoryAuthority.
                AuthorisationDeclined.
                SuggestedChanges.
                ImplementSuggestedChanges.
                ProcessingPersonalData)
                True
            else
                False
        else
            True
    else
        False
else
    False
```

### Article 35 & 37 - Designation and Tasks of the Data Protection Officer

#### Motivation

§1 of article 35 states for what companies and organisations a Data Protection Officer (DPO) are needed. The requirements are for public bodies and authorities, companies and organisations with 250 or more employees or where the core activity of the controller or the processor consists of processing operations which, by virtue of their nature, their scope and/or their purposes, require regular and systematic monitoring of data subjects.

None of these are shown using the Archimate framework and thus there is a need to ask the user whether the Enterprise fits any of these. Proposed is that this is done using the new **GDPR 35** class and add the situations as attributes. If any of the requirements are fulfilled, it is set to true resulting in an easy way of testing the need for a DPO.
Paragraphs two to eleven are all regarding how and when a DPO should be appointed, such as 35.7 stating how long a person should hold the title of DPO and when he or she can be released from this contract. These falls outside of the scope of this thesis since this focuses on checking whether or not the enterprise architecture are compliant with the General Data Protection Regulation.

Article 37 states the tasks of the data protection officer and thus is of interest here. §1 of article 37 states what the general tasks are, such as inform the controller and/or processor of obligations pursuant to the regulation and to monitor the implementation and applications of the policies needed to comply with the regulation.

There are some articles mentioned in §1, meaning that the DPO will have to pay extra attention to them. These articles, 28, 31, 32, 33 and 34, have already been mentioned on pages 31 through 38 above.

The proposed class GDPR 35 will have an attribute to test for §1 of article 35. If any of them are fulfilled there is a need for a DPO. If so, the other paragraphs have to be fulfilled as well.

GDPR 35 is, as all the previous ones, is connected to the GDPR compliance and using this connection, the model will test if there are connections with GDPR 28, GDPR 31, GDPR 32, GDPR 33 and GDPR 34 thus fulfilling the requirements of paragraphs 37.1(d, e and f).

Paragraphs 37.1(g and h) states that the DPO is to be the point of communication with the Supervisory Authority. In order to make sure that there is such a communication, there should, as seen in fig. 42, be a connection between the class GDPRBusinessRoleDPO, a new specialisation of BusinessRole, and GDPRBusinessRoleSupervisoryAuthority using the class Business Collaboration.

Model

![Diagram](attachment:image.png)

**Figure 42:** Proposed model for compliance with article 35 and 37

**Psedu-code**

GDPR\_35:

\[37.1 \{d, e, f\} : \]

\[\text{for } X = \{28, 31, 32, 33, 34\} \text{ do} : \]

\[\text{if } \text{GDPR\_35\_1.GDPRCompliance.GDPR\_X} = \text{True} \]

40
then return True
else return False

37.1\((g,h)\):
if \(\exists\) (GDPR\_35, stakeholder, GDPR\_BusinessRoleDPO, BusinessCollaboration, GDPR\_BusinessRoleSA) then return True
else return False

5.2 Left out articles

Some articles from the General Data Protection Regulation (GDPR) are not discussed in this thesis. The aim of this thesis is to create a tool that can be used in an enterprise environment, thus many articles are left out since they won’t directly affect this goal. For instance, article 56 [10, p.81] is in regards to joint operations of supervisory authorities which by the nature of the article won’t affect an enterprise in a direct way.

It can, however, be argued that some articles left out could be of relevance and therefore they will be discussed below, and the reader will be given the reasons why they have not been included in this thesis.

The overall goal of this thesis is to evaluate and create a tool usable in as many instances as possible. The tool will no deliver a reliable answer if an enterprise is in compliance with the regulation or not, but rather give them a tool to help in the process of verifying that they are.

Article 6 - Lawfulness of processing

Article 6 states the situations where processing of personal data is to be considered lawful. It is vital for an enterprise that any processing of personal data is considered lawful in order to comply with the regulation. However, the reasons for processing varies between enterprises and area of business. Trying to implement this into the model would result in a checklist-like model. So to reduce the size of the model as well as to not create a glorified checklist, this is left out of the model.

Article 7 - Conditions of consent

If falls on the controller to prove that a data subject has given consent to their personal data being processed. This is briefly discussed in article 18 (page 27) but implementing this article in the model would fall outside of the intention of this thesis. In the model, a class correspond to an instance of a unit. So to model that each and every data subject has submitted written consent would require the model to have atleast as many entries as there are data subjects. This would create an extremely complicated model that would be near impossible to use in an efficient manner.

In order to comply with this article, procedures have to be created and reviews of the consent forms have to be performed. Perhaps they can be included in the regular reviews of personal data storages discussed in article 5 (page 18).
Article 8 - Processing of personal data of a child

In the case of an enterprise processing information relating to a person of the age 13 or lower, consent has to be given by the child’s parent or custodian.

The reasoning for leaving this out is much the same as for article 7 (page 41). It would require to model each person 13 years and below as individual units.

Proposed is that, similar to article 7, the control of having consent for these cases are reviewed together with article 5(page 18).

Article 9 - Processing of special categories of personal data

When personal data falls in the categories mentioned in article 9 of the General Data Protection Regulation [10, p.45] any processing is prohibited. However, there are several exceptions to this rule. Testing for each and every one of them would result in a checklist. So to not create such a list within the model, this article is left out. An enterprise still has to ensure that any personal data within these categories are legitimate in accordance with the article.

Article 11 - Transparent information and communication

An enterprise must have any policies of processing accessible in regards to personal data and shall give this information to a data subject upon request. This is very similar to article 14 (page 22) thus this is left out to reduce the amount of double modelling.

Article 20 - Measures based on profiling

Any processing performed in order to evaluate a natural person’s behaviour, performance or the like is prohibited unless at least one of the exceptions mentioned are fulfilled. The author of this thesis believes this is a binary question. Either the processing results in data that is prohibited by this regulation, or it doesn’t. There is no gray zone. The gray zone lies in what is considered profiling, and finding that definition is not within the scope of this thesis. Thus, article 20 is left out of the model but the need for an enterprise to evaluate it is still present.

Article 26 - Processor

According to article 4(6) [10, p.42] the definition of a processor is as follow:

‘processor’ means a natural or legal person, public authority, agency or any other body which process personal data on behalf of the controller;

Article 26 comes in effect if an enterprise outsource the processing of personal data to a third party. For the purpose of this thesis, it is assumed that any enterprise is conducting their own processing of information and thus the need for this article to be implemented in the model is mute.

If this is not the case, an enterprise must take this article into consideration when establishing contact with a processor.

Article 36 - Position of the data protection officer

Article 36 describe the nature of the data protection officer (DPO) position. This is an important part of the DPO. However, it looks more like a job description than something
required in an EA model. Therefore it is proposed that such information is included in the description and/or contract related to the position of DPO.

6 IMPLEMENTATION

The proposed changes have been made using the TOGAF framework as it was intended to be implemented using the Archimate Standard in EAAT [16].

However, Archimate was never meant to be used in reactive modelling, meaning it is to be used for development and testing of proposed changes in a information system. The work in this thesis is to be utilized to verify and keep a system up to date (e.g. article 5, found on page 18). Thus the framework chosen might not be the most suitable.

On top of this, the author has limited knowledge of EAAT and as such, there were many issues when the implementation was to be created. The biggest issue and the main reason that the implementation never became a reality was as follows:

When creating a specialisation of a class in EAAT, the specialisation should be a child of the parent class, as seen in figure 43 where ClassB is a child of ClassA. However, in EAAT what should have become a child class fell on the same level as the parent. The new class became a copy of the original class.

The author tried different approaches to solving this, but none worked. Changing the relations so that the class pointed towards what should be the parent was not allowed (The author do not know why). Creating a class from scratch was also tested but the issue here came with naming variables the same as an already existing class. Since the new class should be a specialisation the same attributes should be present as well as the new ones but since the naming issue was here, this method would not work.

As mentioned, several different methods were tested but none produced a result that fulfilled the TOGAF/Archimate framework while also not making the implementation cluttered. The one method that were implemented so that it functioned required each relation to be named uniquely. So for every specialisation of the Business Process-class that were connected to the Representation-class needed a unique name. This would, as one could imagine, result in a lot of different names meaning the same thing. The test implementation had four such connections. Since a process can both read and write to a representation, the implementation had 8 different relation names corresponding to the two mentioned. Read, Reading and Accessing were all used beside read and it was similar to the write-relation.

In a previous section the proposed changes are presented. Many of them with specialisations. If the above solution would be implemented there would be many relations meaning the same thing but having unique names resulting in a cluttered and messy solution.

Since the authors lack of knowledge of EAAT as well as how to implement TOGAF/Archimate, no working solution was found within the time limit of this thesis.
Figure 43: An image to describe the order of specialisation showing that e.g. ClassB is a child to ClassA same as ClassE is a child to ClassD.

Added on top of this, TOGAF/Archimate were never meant to be a reactive modelling program. It is used to visualize and test proposed architectures, not to test if a system is in compliance with set regulations. As such, there were no date-function implemented in EAAT resulting in some less than perfect programming solution to deal with the time requirement of Article 5 (page 18) where a Julian calendar would be implemented. This results in just working with intervals of 12 months since the authors programming skills are not enough to cover the lack of a built in date handler. So in order to create a functioning implementation, the author would have to put in a considerable amount of work not available during this thesis work and therefore the implementation is lacking. However, the author believes that it is possible to create it and as such, will continue to theorise as if one can be created.

7 EXAMPLE

In order to give the reader a better understanding of how the changes would be implemented, an example model will be presented. In order to keep it simple, a select few of the articles mentioned in section 5.1 (page 16) will be used. Some of the articles are reactive and require companies to have procedures in place to handle them. The example below do not have them and as such, evaluating the architecture on those are not relevant.

7.1 As-is

The architecture used here, as mentioned in section 4 is based on work done for a Swedish company and some information have been censored such as database names as these could potentially reveal the company. What the model depicts, however, is a Business Intelligence solution handling large amounts of personal data. A situation many companies are familiar with and as such it is a perfect fit to exemplify the proposed legislation compliance assessment above.

---

2 Image found at: http://patterns.instantinterfaces.nl/current/Refactoring-and-Design-Patterns-OOP-PRNC-SPC.html
The model in figure 44 describe the work-flow of a Business Intelligence (BI) analyst. It starts with the analyst getting a BI question. Following this the analyst collects the data needed and the model ends when the analyst starts to analyse the data collected from the different systems. The issue that was described during that course is not relevant here. The goal here is to showcase the proposed metamodel changes from this thesis.

As can be seen in figure 44 there are some data nodes in the lower left corner, seen in detail in figure 45. These nodes represent databases containing customer and other personal information used to answer BI questions.

Figure 45: The nodes represent physical databases containing personal information.

Another part of the architecture that will be affected by the proposed changes is the application component used, seen in figure 46.
In the described scenario, the data collection is performed regularly and as such, there are pre-made scripts for different scenarios, shown in figure 47. These are run inside of the application component (figure 46).

In the next section, these three parts of the model will be modified for the proposed changes.

7.2 Modifications for compliance assessment

In this section, the model presented in the previous section will be used to exemplify what companies might have to do in order to utilise the proposed meta-model and to be able to perform a compliance assessment.

The model shown in the previous section requires some modification before compliance with the GDPR can be assessed. To start, let’s look at the model shown in figure 45. The nodes represent physical data storage units running different database software. In order to make it work with the proposed model, these need to be connected to PersonalDataObject classes.

There are two ways of going about this, either the existing model is reworked so that the new classes are incorporated. This means that the nodes are removed and replaced by PersonalDataObjects. The other solution is to extend the current model with the modified metamodel. First solution would require a complete overhaul of the models and possibly the
need to start from the beginning. The second solution would require that for any existing models the new classes are added side-by-side with the current model, e.g. an already existing model of a database containing personal data would require a PersonalDataObject to be added side-by-side to it, as seen in figure 48. However, the second solution could create a more complex model, which goes against the idea of enterprise architecture. The gain from it is that the amount of work required to apply the compliance tests are lowered significantly. For a proof of concept, the second solution is used in this paper.

![Figure 48: One of the databases with the added PersonalDataObject](image)

As mentioned in section 7, only a small part of the proposed metamodel will be used here. Certain articles are not applicable here, such as articles 12 through 19 since they rely on a data subject interacting with the enterprise regarding data related to them. Since the modelled scenario does not contain such interactions, those articles are left out.

The articles that will be modelled here are Article 5 (page 18), Article 23 (page 30) and Article 28 (page 31). Starting with Article 5, all the PersonalDataObjects added will have the attributes related to the article and the dates and review frequency will have to be added to the model. In figure 49 the PersonalDataObjects are connected to the GDPR 5 class to test for compliance with article 5.

![Figure 49: In order to test for compliance with article 5, the PersonalDataObjects has to be connected to the GDPR 5 class.](image)

Article 23 (page 30) is in regard to data protection and who has access to certain data. The article state that the amount of people having access to the system should be finite.
This means, as mentioned above, that there is a requirement of access restriction to the databases. The current architecture (see figure 50) has no such control or if it has, it is not shown in the model.

To remedy this, the new class GDPRAccessRestriction is entered into the model between Search DB and the application component, such that the user is verified and granted access to the database in question. As seen in figure 52, the user has to go through the application function GDPRAccessRestriction to access SAS and the data resulting in a more secure architecture protecting a data subject’s data.

Figure 50: The current solution for database access by the tool SAS.

Figure 51: The modified view of access control for several database interactions
Final article that will be explained here is article 28 (section 5.1) which is in regards to documentation. Each *PersonalDataObject* is required to have a representation connected to it explaining, among other things, what the data is processed for. The required content of the document is listed in the article. Compliance with this article is simple, as can be seen in figure 55 (page 32), since Archimate’s definitions allows us to have a single *Representation* for all the *PersonalDataObjects*. Just connect the class *GDPRProcessingDocumentation* to each of the *PersonalDataObjects* as seen in figure 53.

Finally, in order to test the compliance as mentioned in section 5.1 there is need to connect each of the *PersonalDataObjects* with the corresponding article test classes (*GDPR 5, GDPR 23* and *GDPR 28*). These are then connected to the new class *GDPR Compliance* in order to get a quick overview of compliance.
Figure 54: Each Motivational Requirement specialisation tested here connected to GDPRCompliance for final evaluation of compliance

In this example, the compliance will fail since the example is missing several of the articles mentioned in this thesis.

8 USAGE

As seen in the examples above, installing the proposed changes into an existing model can require some modifications of the model. But once performed, the modified model can be used the same way as before, with the added bonus of having a simple way of testing compliance of the General Data Protection Regulation.

Some changes are simpler than others and some things can work in tandem, such as the above mentioned use of both node and PersonalDataObject.

Other articles require complete rework of how the system is designed, as with compliance of article 23. In some cases the access control is already existent but not modelled, in which case it has to be added to the model. In other cases, the system don’t have access-control implemented in certain systems. In these cases, access control has to be added both to the system as well as the to the models.

Some of the articles, e.g. article 17, require the controller to create procedures and infrastructure to deal with requests done by data subjects. Some enterprises may have such systems in place, some may not.

The author believes that the simplest way of working with the proposed changes would be to look at each article individually, create the model and corresponding organisational and technological changes and after all articles have been implemented, start to rework the model so that the complexity of the system as well as the model is reduced. I.e. work with the new system as with any other but having compliance as one of the main objectives when redesigning the architecture. This is to ensure that each article is handled before trying to reduce the model. However, approaching the issue from another direction could prove useful.

If the requirements are kept in mind when designing or reworking a system, they would be implemented directly without having to worry as much about an unnecessary increase in complexity.
9 DISCUSSION

The work done has resulted in a proof of concept as well as an initial breakdown of the articles from an enterprise architects point of view. Laws and regulations are often written in such a way that they cover as much as possible without allowing loop holes and as such, a person has to interpret the text.

However, since the regulation has just been passed as a legislation and it comes into effect in 2018, there are no precedent cases that can help fine-tune the proposed model.

During the course of this project, the author have been in contact with different law firms in the hopes of having someone more qualified judge the interpretations made. Unfortunately, none of them were available.

9.1 Difference between documents

This thesis has been based on the regulation proposed in 2012 by the European Commission. This document were the first official draft and has since then gone through many versions. Different departments, experts and organisations have been allowed to make their voices heard in regards to the regulation. In 2016, the completed document were presented, and passed, with some changes made to the document.

Some of the old articles have been broken down into several new ones, e.g. article 17 §4 of the old version are now article 19.

Article 31 of the old version stated that after a data breach is detected the supervisory authority has to be informed within 24 hours. This has been changed to now be 72 hours. Since the changes were made official after the main part of this thesis, the changes have not been implemented in this thesis even though some of them changes how the model works. However, this thesis serves as a proof that modelling GDPR compliance assessments are viable and could benefit enterprises.

As mentioned in section 2.2 this thesis and its proposed changes should not be used as a solution for compliance assessment but rather as a tool to very it in conjunction with lawyers and other professionals. There are many free documents published by such law firms to help with understanding the requirements put forth by the new legislation such as the one from Allen & Overy [4] or Bird & Bird [19].

9.2 Possible benefits

The author believes that an enterprise stand to gain several things from implementing the model changes into their existing models. Since, as can be seen in section 4 interpretation of the law text is not straightforward and simple, there will be problems if and when a supervisory authority is involved. With the model, the enterprise have another level of proof that they are compliant with the regulation.

Failing to comply with the regulation could result in expensive fines, up to 4 percent of global turnover for the preceding financial year. For many companies, this would result in huge monetary losses that could be prevented with the proper preparations.

Many of the articles require an enterprise to take action upon requests done by the data subjects. Requests such as removing or changing their data. Such requests could be complicated to comply depending on how a company organises their data.

The requests could also be to transfer data relating to a subject into another, external system. Procedures that require a standardised format not yet set that could change. If they change, enterprises has to modify their system.

As with any modelling a large part of it is simplicity. With a well done model a company gets an overview of their current IT architecture and can validate and test it without
having to utilise the physical structure. In line with this, modelling the GDPR makes it easy to see if the company is in compliance with the legislation.

9.3 Downsides

There are several downsides to having a meta model of the compliance as well. One of them is that meta models are not meant to be used to continuously evaluated a structure but rather to see what the current setup is and test new, modified versions of it. Article 5 (page 18) is the prime example of this since it states that a company must perform regular reviews of their data. The proposed implementation do have such a function, testing the dates of each data storage. Since the models are not meant to be used this way, the review test has do be done in some other system or new procedures has to be set in place to run the meta model test regularly, adding to the amount of procedures already in place.

Secondly, the model is designed in such a way that it fits as many situations as possible, and as such it might not be valid to verify compliance. The model would require updates when the European Commission adds or changes rules in accordance with the regulation. With these updates, the meta model would have to be updated, creating more work for a company.

Thirdly, the legal department is probably the ones responsible for ensuring compliance with the legislation and they are probably not using EA tools to perform their jobs. So in order to benefit from implementing the meta model changes, the legal department would have to work closely with IT possibly creating a need for structural changes. Not to say that such changes might be bad. IT is taking a larger part in enterprises development and with it, a larger portion of the legal work is in relation to IT. But having the legal department working in EA tools require education and training, resulting in costs for a job they might be able to perform without the tool.

9.4 Implementation

The lack of understanding of EAAT made it hard to create a functioning and efficient implementation within a reasonable time period. It is unfortunate but due to lack of time as well as understanding

The implementation could be done in other tools that are designed using Archimate since the proposed changes are all made in accordance with this standard. However, as mentioned above, such an implementation might not be beneficial for enterprises nor is Archimate designed for continuous evaluation.

Perhaps it would be better to give the legislation a few years to settle with the coming changes and modifications and after that redo some of the work performed here so that a more complete model can be proposed.

At that time, an implementation would fit better with any prior cases ruled so that compliance can be ensured using an EA tool. But until then any implementations would require constant modifications resulting, as mentioned, in more time spent updating the model.

9.5 Ethical Aspects

The ethical implications of this thesis are interesting. The subject, privacy, and its ethical implications could generate several thesis’s all on its own. For this project, which is based on a legislation to increase the security of personal data thus ensuring privacy as well as empowering citizens in the EU in cases of data relating to them, the ethical implications are small or non-existent.
The aim was to create a tool that companies and organisations can use to ensure they are compliant with the legislation. By being compliant, they would fulfil the privacy enhancement goals that the legislation aim for. The author can not fantasise a scenario where the work done in this paper can be used in any immoral or unethical ways. As mentioned, the legislation aims to ensure that personal data is handled with care and do have rules against unethical use of personal data. As this paper aim to create a modelling version of the legislation, by extension, also aim to reduce unethical use of personal data.

10 CONCLUSION

This thesis set out to create a meta model implementation of the new General Data Protection Regulation proposed and passed by the EU. The legislation changes how companies have to work with personal data, shifting the burden of proof from the subjects to the companies.

The research done resulted in a proof of concept, a proposed model change based on the Archimate standard for Enterprise Architecture. The proposal has not been implemented into a functioning system. However, the examples given in section 7 shows that modifying existing models can be done, resulting in positive gain as an enterprise gains an overview of their compliance with the legislation.

10.1 Future Work

As mentioned in section 9, the proposed model has not been implemented mainly due to the lack of knowledge of the author. Thus an idea for future work is to create a functioning implementation using the theoretical work done in this thesis.

Also mentioned in section 9 was that currently there are no precedent cases that can be used to fine-tune the proposed implementation. With the coming changes, the model would have to be updated and modified to better fit with the actual outcome of judgements in regards to the legislation.

Furthermore, the proposed should be evaluated together with a lawyer or someone trained in legal work in order to verify that the interpretations of the legal texts are correct.
References


Appendices

A Images

Figure 55: Fullsize image of model used for the example implementation.
Figure 56: Number of leaked records in millions from recent data breaches.